A deep dive into the convolution operation

Romain Thalineau

qwertee.io

romain.thalineau@qwertee.io

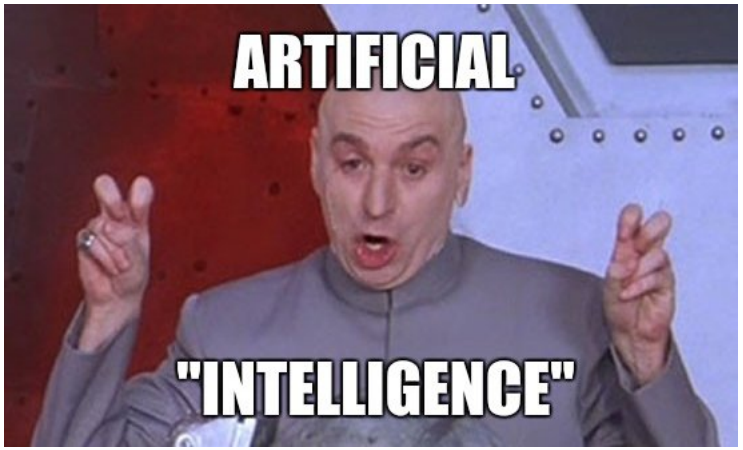
 @romaintha

PyData Cluj-Napoca, November 25, 2019

About

Ph.D. in physics (Grenoble, France)
Quantum computing hardware

Machine learning engineer
Computer vision



Convolutions are everywhere!

Pinned Tweet

Bored Yann LeCun
@boredyannlecun

Many people are not aware that most elementary operations on all numbers (multiplication & addition) are simply special cases of (1x1) convolutions. Let's finally start to teach convolution in its full glory to all children from primary school! #feelthelearn

10:50 PM · Oct 19, 2018 · Twitter Web Client

Intuition

Go to notebook

Convolution: mathematical formalism

The mathematical definition of the convolution of the functions f and g wrtitten $f * g$ is:

$$s(t) = (f * g)(t) = \int_{-\infty}^{-\infty} f(\tau)g(t-\tau)d\tau$$

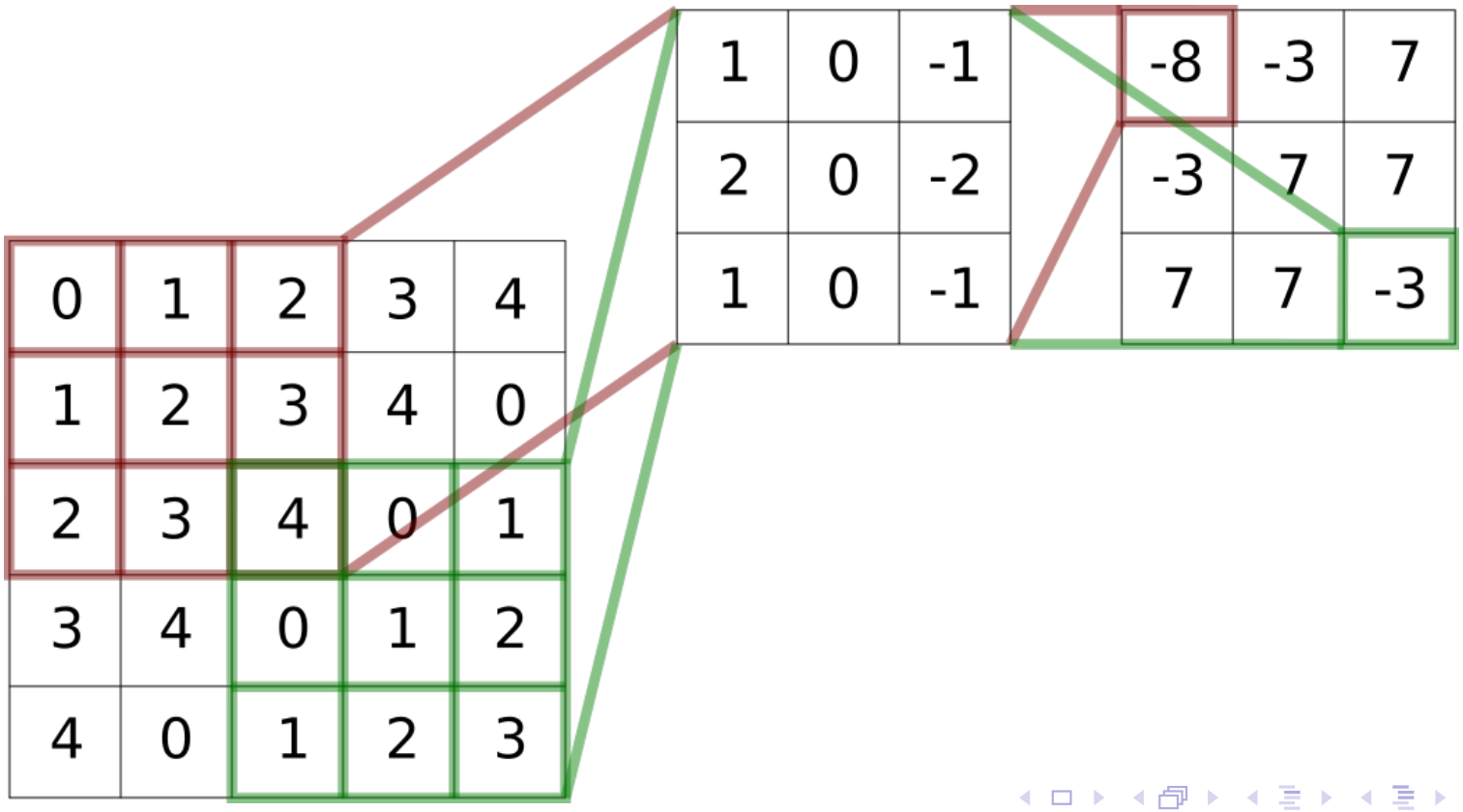
In the discrete case, the convolution is expressed as:

$$s(t) = (f * g)(t) = \sum_{n=-\infty}^{-\infty} f(n)g(t-n)$$

From 1d to 2d convolutions

Considering an image I and a kernel(filter) K, both being 2 dimensional, the convolution S of I and K is:

$$S(i,j) = (I * K)(i,j) = \sum_{m=-\infty}^{-\infty} \sum_{n=-\infty}^{-\infty} I(m,n)K(i-m, j-n)$$



Convolutions for image processing

Go to notebook

Image recognition

Learn the conditional probability $p(y|x)$ where:

y: muffin or chihuahua

x: input image

Deep learning: Universal Approximation Theorem

Universal Approximation Theorem[1]:
a MLP can approximate any function with any degree of accuracy (given enough hidden units)

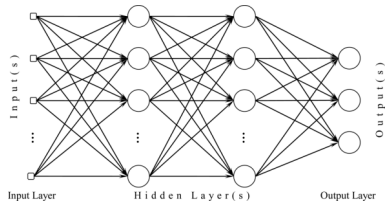
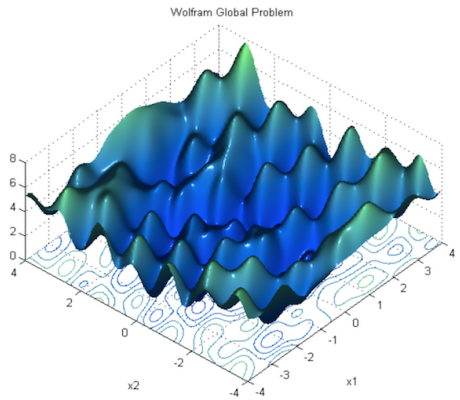


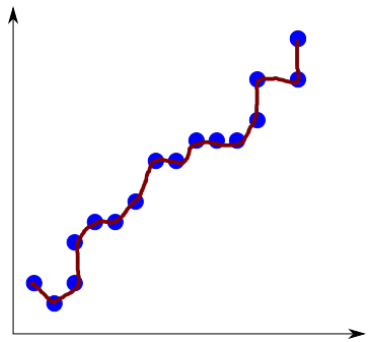
Image source: http://www.mdpi.com/

But:

No guarantee of learning it

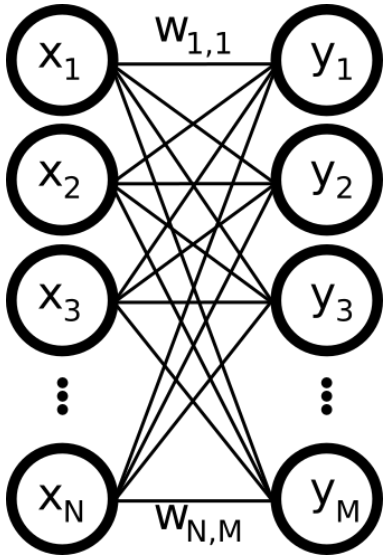


Number of hidden units required $\propto e^{input\_size} \rightarrow$ Overfitting



[1] Hornik et al (1989)

Deep learning: MLP



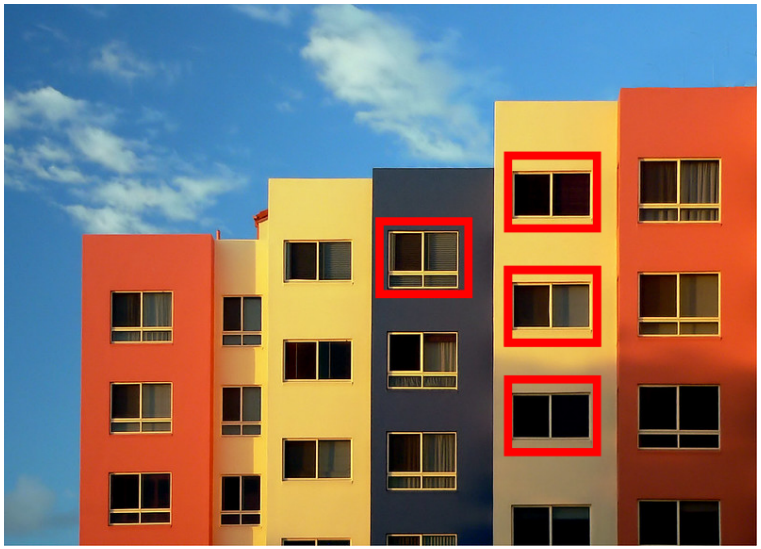
Any output y_i is simply a linear combination of all inputs x_i to which an activation function σ is applied:

$$y_i = \sigma(b_i + \sum_{j=1}^{N} w_{j,i} x_j)$$

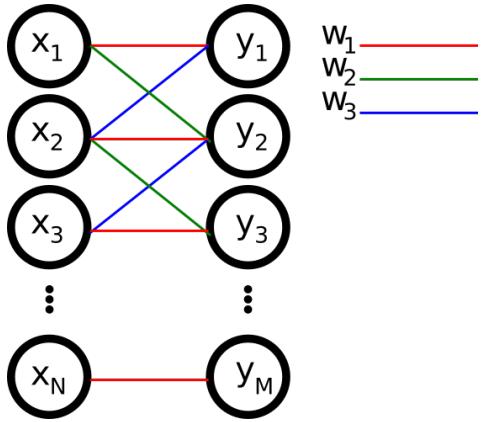
Image key properties

Locality

Translation invariance



Deep learning: convolution



The output y_i can be written as a linear combination of only a few input x_i:

$$y_i = \sigma(b_i + w_1 x_i + w_2 x_{i-1} + w_3 x_{i+1})$$

This leverages 3 important ideas: **sparse connectivity**, **parameter sharing**, **equivariance to translation**

Deep learning: convolution arithmetic

$$o = [\frac{i + 2p - k}{s}] + 1$$

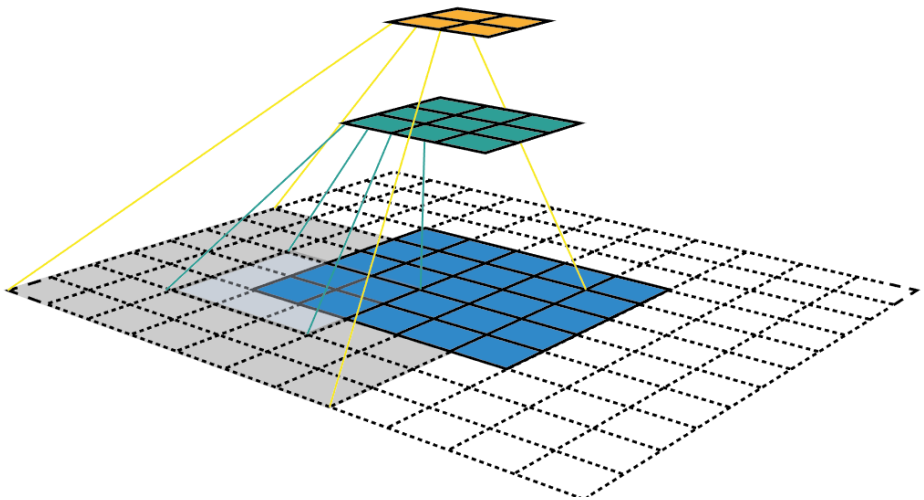
where:

- o: output size
- i: input size
- p: zero padding (number of zeros added around the image)
- k: kernel size
- s: stride (distance between two consecutive positions of the kernel)

More on convolution arithmetic here.

Deep learning: deep convolutional network

Stacking simple blocks allows for learning higher level features [1]



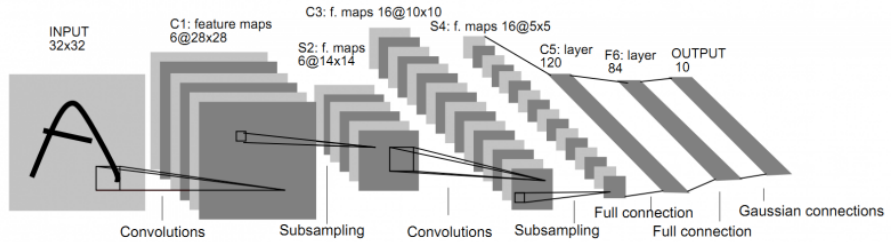


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

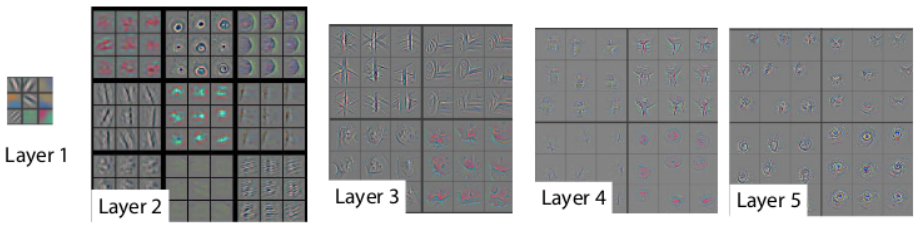
[1] Lecun et al

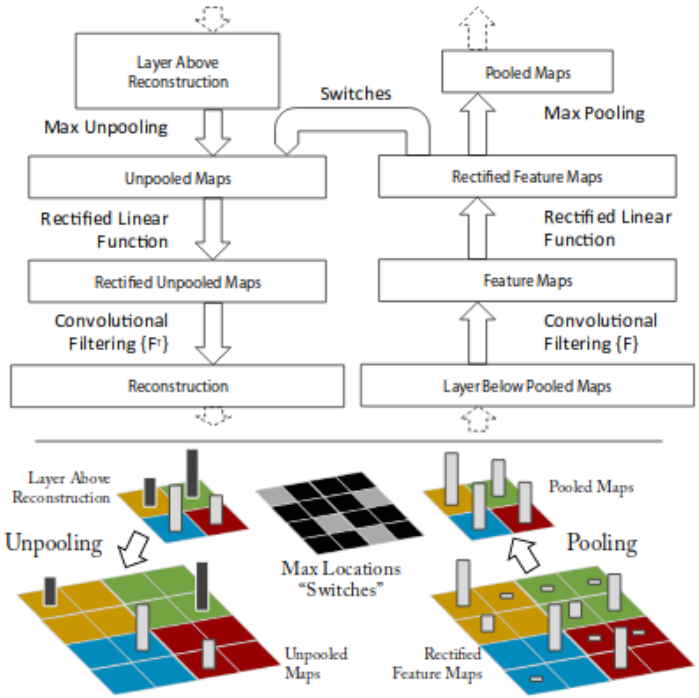
Deep learning: visualizing CNN

Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

Rob Fergus FERGUS@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

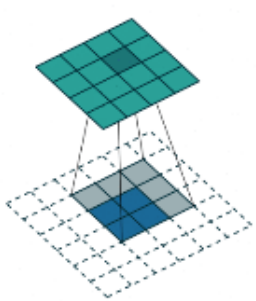




Go to notebook

Deep learning: other kind of convolutions

Transposed convolution
(a.k.a. deconvolution)



Dilated convolution
(a.k.a. atrous convolution)

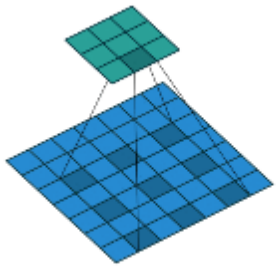


Image by Vincent Dumoulin

Extra resources

- Deep learning book
- Dive into deep learning book
- Understanding convolutions
- Feature visualization
- Deconvolution and checkerboard artifacts

Thank you for your attention